# Efficient Inference with Multiple Heterogeneous Part Detectors for Human Pose Estimation

Vivek Kumar Singh, Ram Nevatia, and Chang Huang

University of Southern California, Los Angeles, USA
{viveksin,nevatia,huangcha}@usc.edu

**Abstract.** We address the problem of estimating human pose in a single image using a part based approach. Pose accuracy is directly affected by the accuracy of the part detectors but more accurate detectors are likely to be also more computationally expensive. We propose to use multiple, heterogeneous part detectors with varying accuracy and computation requirements, ordered in a hierarchy, to achieve more accurate and efficient pose estimation. For inference, we propose an algorithm to localize articulated objects by exploiting an ordered hierarchy of detectors with increasing accuracy. The inference uses branch and bound method to search for each part and use kinematics from neighboring parts to guide the branching behavior and compute bounds on the best part estimate. We demonstrate our approach on a publicly available People dataset and outperform the state-of-art methods. Our inference is 3 times faster than one based on using a single, highly accurate detector.

**Keywords:** Human pose estimation, part based models, branch and bound, message passing.

## 1 Introduction

We consider the problem of localizing 3-D articulated objects such as humans in their 2-D images; the projected shape varies with the viewpoint and articulations, we choose to model these variations as deformations. An intuitive and widely accepted approach to model an articulated object is to decompose the object into smaller objects (parts) and model the deformability by loose spatial relationships between the parts. [1], [2], [3], [4], [5] used such part based representations to detect and localize objects with large variations. The localization accuracy increases with better part detectors but it comes at the cost of increased computation. We enhance the part based model with multiple heterogeneous features for better detection accuracy, and propose a novel progressive search based method for efficient inference.

A common model for part based object representation is that of *pictorial structures*, which is a tree-structured graphical model that represents the kinematic relationships between the parts; pose is inferred by enforcing kinematics constraints on part hypotheses that are obtained by applying part detectors. Such part based approaches can be *dense* or *sparse* based on how parts are sampled from the image. *Dense sampling* methods [1], [6], [7], [8] apply each part

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part III, LNCS 6313, pp. 314–327, 2010.

detector over all possible locations, orientations and scales; [1] presents exact and efficient inference on densely sampled parts, however for better efficiency, these methods tend to use part detectors that are generally weak. [7] shows that better part detectors can significantly improve the performance accuracy; however better part detectors are often more complex and computationally expensive. The *sparse sampling* methods approximate the part likelihood using few hypotheses and infer the pose from these hypotheses [9], [10], [2], [11]. To avoid applying the part detector over the entire space, these approaches obtain part hypotheses from bottom up feature responses such as by using parallel line segments [10], [2]. For inference, [9] uses non parametric belief propagation which is slow due to its stochastic nature; [10], [11] use integer programming methods to infer pose, but the size of the program grows rapidly with increase in the number of candidate part hypotheses.

An alternate representation is to use hierarchical model in which multiple levels represent the object at varying granularity [3], [12], [5]; parts need not correspond to the natural object parts (such as limbs for human). [13] presented a generic AND/OR graphical model for deformable objects, where the leaf nodes are points on the boundary and the intermediate nodes represent different object parts. [14], [15], [16] use 2-level hierarchical models to find humans with a whole object representation at higher level and subsequent parts at the lower level. [16] used Poselets (tightly coupled local part configurations) at the lower level to achieve more accurate detection. However, these methods use pose-restrictive assumptions such as upright human where torso is above the legs.

We use a densely part sampled approach in this work and propose to use multiple heterogeneous detectors for each part to achieve a higher detection accuracy. More precisely, we use a linearly weighted combination of multiple detectors for a part, and order the detectors by their discriminability and efficiency. The combination weights are learnt in a discriminative framework using *Voted Perceptron* [17], so that the combined detector has better accuracy than the individual detectors. Further, the ordering of the detectors is selected such that as we go up the order, the part detectors become more accurate and precise, but also computationally more expensive.

For efficient inference over the graphical model with ordered heterogeneous features, we propose a novel *collaborative branch and bound algorithm*. The key idea is to use branch and bound search for each part, where the bounds on the best part estimate are obtained by enforcing kinematic consistency between the search branches of neighboring parts; thus, the kinematic constraints form a collaboration model between part search branches. At each step, search space is reduced by applying a more accurate detector for each part and pruning out the subsets that are less likely to contain the best part estimate. The best part estimate refers to the estimate otherwise obtained by dense sampling parts using all the detectors, which is highly inefficient. We demonstrate our approach on a commonly used dataset of complex human poses in cluttered backgrounds [6]. Our algorithm gives better than the state-or-art accuracy on the dataset and inference is $\sim 3$ times faster than one using a single, highly accurate detector.

In the rest of the paper, we first discuss the pictorial structure with multiple part detectors in section 2 and the collaborative branch and bound algorithm in section 3. Next, we present the parameter learning in section 4, followed by the experiments in section 5 and conclusion in section 6.

## 2 Pictorial Structure with Multiple Part Detectors

We use pictorial structures [1] to model humans. Instead of part detectors with one type of feature descriptor such as shape context [7] or Haar-like [6], we use detectors with multiple heterogeneous features for each part. We impose a hierarchical ordering on these detectors such that the model becomes more precise as we go up the hierarchy. Thus, different levels in the hierarchy represent part detectors with different descriptors. Figure 1 shows the pictorial structure for full body and a 3-part model with $n$ detectors for each part.

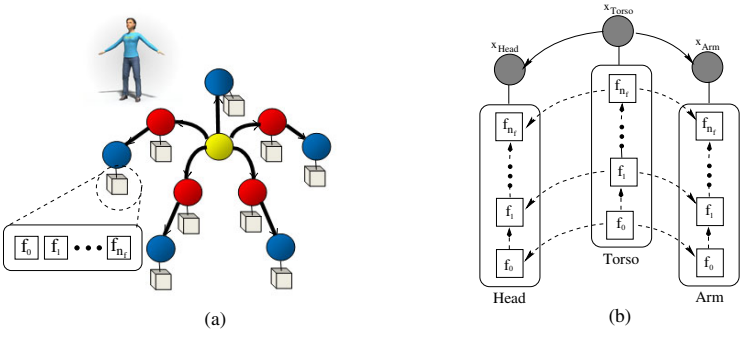


**Fig. 1.** Pose Model: (a) Tree structured human model, with one of the observation nodes showing multiple detectors (b) A 3 part model showing detectors in a hierarchical order. Note that the dotted arrows are not graphical model links but are shown to indicate the relationship between the different detectors.

**[Representation Details]**
We assume that the output of different detectors of a part are independent of each other and that the part likelihood distributions are conditionally independent given the full object $\mathbf{x}$. Under these assumptions, the posterior log-likelihood of the object $\mathbf{x}$ obtained by applying detectors at level $k$ is given by,

$$\mathcal{F}^k(\mathbf{x}, I) = \sum_{i \in V} \phi_i^k(I|x_i) + \sum_{ij \in E} \psi_{ij}^k(x_i, x_j) \quad (1)$$

where $(V, E)$ is the graphical model, $I$ is the image observation, $\phi_i^k$ is the likelihood for part $i$ using detector $f_k$; we refer to $\phi_i^k$ as the *part support* for $i$ using $f_k$. The log-likelihood in equation 1 is same as log-likelihood function of a pictorial structure. Note for simplicity we assume that the kinematic models are same at all the levels *i.e.* $\psi_{ij}^k = \psi_{ij}$ for all $k$, and are assumed to be Gaussian.

Given the part supports $\phi_i^k$s obtained by applying all the detectors for part $i$, we combine them to obtain the part likelihood distribution. Since the accuracy of detectors for each part may vary with parts, we associate a weight with all the detectors for every part. We then define the optimal pose configuration $\mathbf{x}^*$ by

$$\mathbf{x}^* = arg\max_{\mathbf{x}\in\mathbf{X}} \left( \sum_{i\in V} \sum_{k=1}^{n_f} w_i^k \phi_i^k(I|x_i) + \sum_{ij\in E} \psi_{ij}(x_i, x_j) \right) \tag{2}$$

where, $w_i^k$ is the weight associated detector $f_k$ for part $i$. Our part representation with hierarchical feature model (eqn 2) can be interpreted as a combination of $n_f$ tree pictorial structures, each with part detectors with different features.

**[Model Constraints]**
For selection of part detectors that are useful for efficient and more accurate inference, we impose the following constraints on the set of detectors used for each part,
$\triangleright$ For the ground truth pose $\mathbf{x}^{gt}$, $\mathcal{F}^{k+1}(\mathbf{x}^{gt}, I) > \mathcal{F}^k(\mathbf{x}^{gt}, I)$, where $\mathcal{F}^k(\mathbf{x}, I)$ is the log-likelihood obtained using features at level $k$ (see eqn 1). This ensures that the detectors at level $k+1$ have better localization accuracy than the detectors at a lower level.
$\triangleright$ For each part $i$, $time(\phi_i^{k+1}) > time(\phi_i^k)$ *i.e.* the detectors at higher level are less efficient than the detectors at lower level.

## 3 Inference

Next we describe our collaborative search approach for efficient inference over the pictorial structures with hierarchical feature ordering. [1] proposed efficient algorithms for inference on pictorial structures using sum-product/message passing algorithm. However, the proposed method uses a dense search for each object part and hence becomes inefficient with complex (more discriminative) part detectors. [18] proposed an efficient algorithm for localization using complex features by defining quality functions that bound the probability of finding the object within a window. In this work, we use the branch and bound algorithm to iteratively search for the object part using coarse-to-fine features. However in order to quickly concentrate the search in high likelihood regions, we make inter-branch inferences between the active search branches. In other words, the branching algorithm depends on the estimated likelihood of the neighboring parts in the current iteration.

We proceed by briefly describing the message passing algorithm, and then introduce our inference algorithm.

**[Belief Propagation on Pictorial Structures]** [19]
Pictorial structures infers the object position by maximizing joint distribution of the parts (see eqn 1). [1] presented efficient algorithms for inference using

belief propagation [19]. The algorithm simultaneously computes the posterior distribution of all parts by locally exchanging *messages* between connected parts. The message from part $i$ to part $j$ is the distribution of the joint connecting parts $i$ and $j$, based on the observation at part $i$. This distribution is efficiently obtained by transforming the part distribution into the coordinate system of the connecting joint (using eqn 3) and applying a zero mean Gaussian whose variance determines the stiffness between the parts.

$$T_{ij}(x_i) = \begin{pmatrix} px_i + s_i\mu_x^{ji}cos\theta_i - s_i\mu_y^{ji}sin\theta_i \\ py_i + s_i\mu_x^{ji}sin\theta_i + s_i\mu_y^{ji}cos\theta_i \\ \theta_i + \mu_\theta^{ji} \\ s_i \end{pmatrix} \tag{3}$$

where, $x_i = (px_i, py_i, \theta_i, s_i)$ is position, orientation and scale of $x_i$, $(\mu_x^{ji}, \mu_y^{ji})$ is the mean relative position of the joint between $i$ and $j$, and $\mu_\theta^{ji}$ is the relative angle of part $j$ in the coordinate frame of part $i$.

### 3.1 Collaborative Branch and Bound Algorithm

Now we describe the collaborative search algorithm for a tree-structured pictorial structures with hierarchical part features. We introduce the following terms and notations for our algorithm. The *configuration space* of part $i$ is denoted with $D_i$; the response function for each part detector is assumed to be of the form $\phi_i : R^d \rightarrow R$. Instead of working over the entire configuration space for each part, the algorithm maintains subsets that are most likely to contain a true part response. We refer to these subsets as *active*. As the subsets are obtained by branching, each active subset is associated with a unique *active branch*, and for each part $i$, the current set of active branches is denoted by $\mathcal{R}_i$. The subset spanned by a branch $t_i$ is denoted as $S(t_i)$.

**[Initialization]:** Initiate the search for each part $i$ over the entire domain space $D_i$. Thus the initial set of *active branches* $\mathcal{R}_i = D_i$.

**[Iterative Step]:** Given sets of active branches for all parts $\mathbf{R}$ at the current iteration $k$, apply the next detector $f_k$ over all branches $\in \mathbf{R}$. Note that as the new detector is applied, the part support at each part location gets accumulated *i.e.* value at location $l$ after applying $k^{th}$ detector $f_k$ is $\sum_k w_k\phi^k(l)$.

*Branching*: For each branch $t_i$, uniformly partition the space $S(t_i)$ into non-overlapping subsets $\{S(t_{i,c})\}$ such that $S(t_i) = \bigcup_c S(t_{i,c})$, and initialize new branches $t_{i,c}$ over each partition $S(t_{i,c})$ and add them to active branch set $\mathcal{R}_i$. For simplicity, each space is partitioned uniformly along the all dimensions and thus all the subsets are blocks (hypercuboids). For each active branch including the new branches, say $t_i$, compute the *max*, *min* and *average* of the detection responses over the subset spanned by the branch $S(t_{i,c})$, denoted by $max(t_i)$, $min(t_i)$ and $avg(t_i)$ respectively. We refer to these values as *branch statistics*.

*Inter-branch Message Passing*: We use inter branch message passing to compute the quality $q(t_i)$ of each branch $t_i$. For inter-branch message passing, we define the joint likelihood between a pair of branches from connected parts using eqn 4,

$$\psi(t_i|t_j) = -logN(\delta(T^{ji}(t_i), T^{ij}(t_j)); \mu^{ji}, \Sigma^{ji}) \tag{4}$$

Here, $T^{ji}(t_i)$ is the transformed subset of $t_i$ in the coordinate system of the joint between parts $i$ and $j$ (given by eqn 3), and $\delta(t_i, t_j)$ is the minimum displacement vector between the subsets spanned by $t_i$ and $t_j$ *i.e.* $S(t_i)$ and $S(t_j)$. More precisely,

$$\delta(t_i, t_j) = \mathbf{l_i} - \mathbf{l_j} \quad \textit{where} \quad (\mathbf{l_i}, \mathbf{l_j}) = \arg\min_{\mathbf{l_i} \in S(t_i), \mathbf{l_j} \in S(t_j)} ||\mathbf{l_i} - \mathbf{l_j}||^2$$

Note that since each subset is a block, the minimum displacement vector can be computed very efficiently, and is independent of the size of the subset.

We now define the quality $q(t_i)$ of a thread $t_i$ as the joint posterior of average branch statistic of $t_i$ and max statistic of all threads $t \neq t_i$.

$$q(t_i) = avg(t_i) + \max_{t_i \cup (\mathbf{R} \setminus \mathcal{R}_i)} \left( \sum_{j \in V, j \neq i} max(t_j) + \sum_{(j,k) \in E} \psi(t_j|t_k) \right) \tag{5}$$

This definition of the branch quality roughly captures the idea that higher branch quality implies greater likelihood of the best part hypothesis to belong to the subset spanned by that branch. Note however that since quality is defined on average statistic, there is no guarantee that the highest quality thread will always have the best part hypothesis.

Compute the bounds on the branch quality $q(t_i)$ using inter-branch message passing over with max and min branch statistic of $t_i$. More precisely, to obtain an upper bound on $q(t_i)$, say $q^U(t_i)$, compute the joint posterior of max branch statistic of $t_i$ and all threads $t \neq t_i$. Similarly, compute the lower bound $q^L(t_i)$ as the joint posterior of min branch statistic of $t_i$ and min statistic of all threads $t \neq t_i$. Note that equation for $q^U(t_i)$ and $q^L(t_i)$ can be written by replacing the first term in eqn 5 by $max(t_i)$ and $min(t_i)$ respectively.

*Pruning*: Clearly, the quality of branch $q(t_i)$ belongs to $[q^L(t_i), q^U(t_i)]$. More importantly, notice that $q^L(t_i)$ and $q^U(t_i)$ are upper and lower bounds of the best part estimates for all $x_i \in S(t_i)$ (best part estimate is referred to best hypothesis obtained by belief propagation over entire part space). After these bounds are computed for all the active branches in $\mathcal{R}$, prune the branches in $t_i \in R_i$ when $q^U(t_i)$ is lower than the lower bound of another branch $\hat{t}_i \in R_i$ *i.e.* $q^U(t_i) < q^L(\hat{t}_i)$. Note that since we are pruning based on the accumulated part support upto level $k$, the pruning is not guaranteed to retain the best hypothesis for each part. However, in our experiments, we did not observe any loss in accuracy. For greater efficiency, we also prune the branches with a very low quality.

After pruning we then apply the next detector and repeat until all the detectors have been applied.

**[Selecting the Best Pose]:** After we have applied all the features by progressively reducing the search space, we still need to estimate the best pose over the current set of active subsets. One may continue to use branch and bound to reduce the search space, however as branches increase the computational overhead for computing the branch bounds increases. So we use belief propagation over the active subsets to obtain the part posterior distribution, and best pose is obtained by assembling the MAP for each part.

The search algorithm is summarized in Algorithm 1.

**Algorithm 1.** Collaborative Branch and Bound Algorithm

Initialize each $\mathcal{R}_i$ with a single thread over $D_i$
**for** $k \leftarrow 1$ to $|\mathcal{F}|$ **do**
  $\triangleright$ For each branch $t \in \mathbf{R}$, apply the next detector $f^k$ over the entire subset $S(t)$
  *Branching*
  **for** each part $i$ **do**
    $\triangleright$ $\mathcal{R}_i \leftarrow partition(\mathcal{R}_i)$
    $\triangleright$ Compute $max$, $min$, $average$ statistic for each branch $t \in \mathcal{R}_i$
  **end for**
  *Inter-thread Message Passing* {compute the quality of each active thread}
  $\triangleright$ Compute quality $q(t)$ for each branch $t \in \mathbf{R}$ using equation 5
  $\triangleright$ Compute quality bounds $q^L(t)$ and $q^U(t)$ for each branch $t \in \mathbf{R}$
  *Pruning*
  **for** each part $i$ **do**
    $\triangleright$ Remove branch $t$ if there exists $t_k \in \mathcal{R}_\rangle$ $q^U(t) < q^L(t_k)$
  **end for**
**end for**
$\triangleright$ Compute joint posterior distribution of the parts over the active search space
$\triangleright$ Obtain the best pose by collecting MAP estimate of each part

**[Accuracy vs Efficiency Tradeoff]:** The key aspect of our inference is that the branching and pruning step depends not on the detection responses but also on the kinematic constraints. This allows the algorithm to quickly focus on the subsets that are most likely to contain the best hypothesis. The efficiency of the algorithm depends on branching factor. When branching factor is too high, the inter-branch message passing becomes computationally expensive since the overhead of computing bounds becomes significant. When the branching factor is small, higher level detectors will be applied over a large space, thereby slowing down the inference. Since optimal branching factor for highest efficiency depends on the accuracy of detectors on the input image itself, we empirically select the branching factor based on the performance of our algorithm on the training data.

## 4 Parameter Learning

Model parameters include the kinematic functions $\psi_{ij}$s and model weights $\{w_i^k\}$.

**[Kinematic Prior]:** The kinematic function is modeled with Gaussians, *i.e.* position of the connecting joint in a coordinate system of both parts $(m^{ij}, \sigma^{ij})$ and

$(m^{ji}, \sigma^{ji})$ and the relative angles of the parts at the connected joint $(m_\theta^{ij}, \sigma_\theta^{ij})$. Given the joint annotations that is available from the training data, we learn the Gaussian parameters with a Maximum Likelihood Estimator [1], [7].

**[Feature Weight Vector]:** Since the joint likelihood function is log-linear (eqn 2), we learn the model weights using the *Voted Perceptron algorithm* [17]. The algorithm computes the prediction error given the current set of weights and update the weights based on the error between the true and predicted positions (see Algorithm 2). The algorithm doesn't converge to a zero error but rather gets close a certain value and oscillates around it. Thus as in [17], we use the weights obtained by averaging weights obtained over many runs.

**Algorithm 2.** Model Weight Learning using Structured Perceptron

Randomly set the initial weights $\overline{w}$
**for** $t = 1$ to $T$, $j = 1$ to $N$ **do**
  ▷ Compute MAP pose $\mathbf{x}^o$ on training image $\mathcal{I}^j$ using current weights $\overline{w}$.
  **for** $p = 1$ to $n$ **do**
    **if** $x_p^o \neq x_p^{gt}$ **then**
      ▷ Collect the feature error vector, $\Delta_p^k = \Delta_p^k + \phi_p^k(\mathcal{I}^j|\mathbf{x}^{gt}) - \phi_p^k(\mathcal{I}^j|\mathbf{x}^o)$
      ▷ Update the weight vector, $w_p^k = w_p^k + \frac{\Delta_p^k}{||\Delta_p||_{L^1}}$
    **end if**
  **end for**
**end for**

## 5 Experiments

For evaluation, we use the People dataset [6] which contains 305 images, each with highly articulated human poses in a cluttered background. The images are resized such that the human in these images are about 100 pixels high. The first 100 images were used for training and rest 205 images used for testing, as in other reported experiments on this dataset [6], [7].

### 5.1 Part Detectors

We use 3 detectors in a hierarchy for each part - minimum edge density (fast but low accuracy), boundary and region templates (slower but more accurate) and boosted cascades (slowest but high accuracy). Figure 2 shows the features and templates used in part detectors.

**[Edge Density Filters]:** We learn edge density filters for each body part from the training set. Edge density filters (EDF) find the potential candidate regions for each part based on the density of edges within a window. This involves computing a Sobel edge map, followed by thresholding applied at each location based on the density of edges within the window. For each part, a square window of

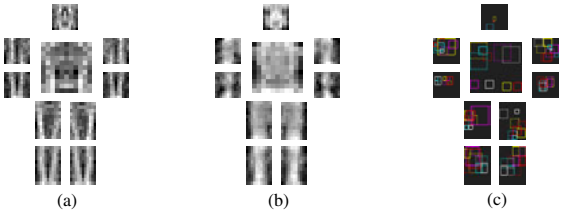


**Fig. 2.** Templates and features used in part detectors: (a) Boundary templates (b) Region Templates, dark areas correspond to low probability of edge, and bright areas correspond to a high probability; (c) Selected JRoGs, each granule-pair used to compute the feature is indicated by the same color. For clarity, only the first few pairs are shown for each part;

size equal to the length of the part was considered, and integral images were used for efficiency. Note that this filtering process mainly helps in removing obvious regions where the density of edges is low.

**[Boundary and Region Templates]:** We used the boundary and region templates trained by Ramanan et al [6] for localizing human pose (see 2(a, b)). Each template is a weighted sum of the oriented bar filters where the weights are obtained by maximizing the conditional joint likelihood (refer [4] for details on training). The likelihood of a part is obtained by convolving the part boundary template with the Sobel edge map, and the part region template with part's appearance likelihood map. Since the appearance of parts is not known at the start, part estimates inferred using boundary templates are used to build the part appearance models. For each part, an RGB histogram of the part $h_{fg}$ and its background $h_{bg}$ is learnt; the appearance likelihood map for the part is then simply given by the binary map $p(H_{fg}|c) > p(H_{bg}|c)$. For more details, please refer to [6].

**[Boosted JRoG Cascades]:** We trained discriminative part detectors using the JRoG (Joint Ranking of Granules) proposed in [20]. A JRoG captures region dissimilarity between a pair of image blocks due to the difference in grayscale value or the average gradient, and is more accurate and efficient on pedestrian detection tasks [20] than other popular features such as HOG [21] and Edgelets [22]. The likelihood of a part hypothesis is given by the sum of the fraction of layers passed (in the cascade) and detection confidence obtained by the final layer (if all layers are passed). We trained each part cascade detector using boosting with 5000 positive training examples for each part and a million negative examples from the images obtained from the Internet. The positive samples were obtained by the 100 part annotations, their flipped versions and small affine perturbations (rotation, scale and translation). Figure 2(c) shows the first selected JRoG features.

### 5.2 Human Pose Inference

We used the following hierarchy of detectors: edge-density filters, boundary templates, boosted JRoG cascades and region templates. The joint posterior distribution of parts was efficiently computed using the proposed collaborative branch and bound algorithm. We used RGB histograms discretized into $16 \times 16 \times 16$ bins to model appearance of each part. Part appearance models were learnt using the joint part posterior of the boundary templates and the boosted JRoG cascades. Note that unlike [6], we learn part appearance models and apply the region templates only over the active search space (after applying the JRoG detector) which is much smaller than entire part space.

### 5.3 Evaluation

We compute the pose accuracy of the entire dataset by computing the average correctness of each body part over all the images (total of 2050 parts). An estimated body part is considered correct if its segment endpoints lie within 50% of the length of the ground-truth segment from their annotated location, as in earlier reported results [7].

**[Part Detectors]:** We compare the performance of the part detectors used in this work over the entire test set. Each part detector was applied at every pixel in 24 orientations. Both accuracy and timing of the part detectors trained with different features are shown in Table 2. We observed that even though the person in each image has been scaled to approximately 100 pixels, the variance in part sizes is still significant (*e.g.* lower leg has a standard deviation of $\sim 10$ pixels over the entire dataset). So we applied the detectors over 5 scales - $\{0.8, 0.9, 1.0, 1.1, 1.2\}$. While applying template detectors over multiple scales did not improve the detection accuracy, the boosted JRoG part detectors applied over multiple scales was much more accurate than for a single scale (shown in Table 1). Thus, in our experiments we apply the template detector over a single scale and the JRoG detectors over all 5 scales. Also notice that the appearance models obtained from more accurate detection responses, improve the accuracy of the region templates [6]. Thus, we use the region templates at the top of our hierarchy.

The times reported in Table 2 are the total time taken by densely applying all the detectors to process a $276 \times 213$ image. Timing is computed on a state-of-art machine with 3GHz Dual-Quad Core CPU. Since the part detectors are independent of each other, we apply part detectors in parallel using OpenMP programming. This speeds up the detectors by about $\sim 5$ times. Template detectors were applied using the fast convolution method in the Intel Image Processing Primitives library.

**[Pose Estimation]:** We evaluate the performance of our system at various levels in the hierarchy with different combination of part detectors.

*Accuracy*: Table 2 shows the pose estimation accuracy averaged over all parts (total of 2050 parts). Notice that the detection accuracy increases with the use of

**Table 1.** Comparison of Part Detector accuracy. Arm and leg numbers are average of left and right parts. The appearance models for region detectors in row 3 and 4 are obtained by using responses from boundary templates and JRoG detectors (shown in parentheses).

| Method | Accuracy | | | | | | | Time |
|---|---|---|---|---|---|---|---|---|
| | Torso | U-Leg | L-Leg | U-Arm | L-Arm | Head | Total | (in secs) |
| Edge Density Filters | - | - | - | - | - | - | - | 0.12s |
| Boundary Templates | 3.9 | 6.3 | 9.7 | 2.4 | 5.6 | 3.9 | 5.61 | 2.032s |
| Region (wBoundary) | 18.0 | 13.4 | 10.0 | 1.9 | 1.7 | 0.0 | 7.22 | 2.047s |
| Region (wBoundary+JRoG) | 30.7 | 26.6 | 14.8 | 3.9 | 1.7 | 1.9 | 12.68 | 2.047s |
| JRoG | 28.8 | 16.5 | 5.8 | 3.4 | 1.0 | 32.4 | 11.48 | 2.640s |
| JRoG (5 scales) | 34.1 | 23.1 | 13.2 | 5.1 | 5.8 | 42.4 | 17.13 | 13.843s |

more heterogeneous detectors for each part. Also note that our selection of part detector ordering clearly satisfies the model constraints (increasing accuracy and computation time with increase in level) given in Section 2.2. Learning of feature weights increases the performance accuracy by about $\sim 2\%$. Figure 3(a) shows the variation in the number of poses with number of detected parts per pose; notice that with increase in number of detectors, the number of poses with fewer detected parts decreases while the number of poses with higher detected parts increases.

*Timing*: The speed up factor using our algorithm depends on the amount of pruning achieved by the faster part detectors. Hence in images with significant background clutter the speed up factor is smaller. Table 2 shows the time taken to process a $\sim 200 \times 180$ image averaged over 50 images from the People dataset. Our inference is about 3.6 times faster than Pictorial Structures' dense search [1]. Over the entire dataset, the speed up factor varied between 3 to 5 with an average factor of $\sim 3.7$. When the detectors were applied sequentially (without parallel programming), the average speed up factor increased to $\sim 4$. Figure 3(b) illustrates the progressive search space reduction at various levels in the hierarchy over the entire dataset.

**Table 2.** Pose Estimation accuracy with different part detector combinations. Time reported is averaged over 50 images of size $\sim 200 \times 180$.

| | Accuracy | Time (in secs) | |
|---|---|---|---|
| | | Dense Parsing [1] | Collaborative BnB |
| EDF + Edge Templates | 33.95 | 1.785s | 1.487s |
| EDF + Edge + Region Templates | 43.90 | 3.711s | 2.987s |
| EDF + Boosted JRoG Cascades | 51.95 | 8.727s | 6.807s |
| EDF + Edge + JRoG | 54.78 | 9.832s | 3.096s |
| EDF + Edge + JRoG + Region | **60.88** | 11.76s | 3.27s |

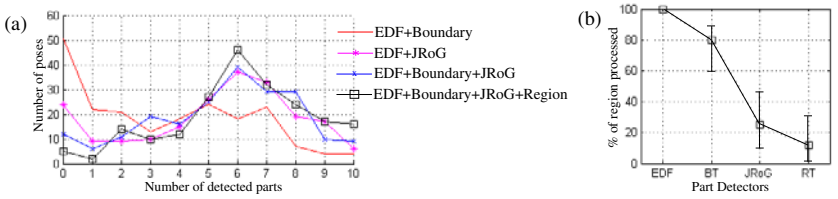


**Fig. 3.** (a) Number of poses vs number of detected parts per pose (b) the percentage of configuration space processed by detectors at different levels in hierarchy

**Table 3.** Comparison of Pose Estimation Results with other approaches: Iterative Image Parsing (IIP) [6], Boosted Shape Context with Pictorial Structures (PS) [7].

| Method | Torso | Upper leg | | Lower leg | | Upper arm | | Forearm | | Head | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Boundary-PS [6] | 39.5 | 21.4 | 20 | 23.9 | 17.5 | 13.6 | 11.7 | 12.1 | 11.2 | 21.4 | 19.2 |
| Boundary+Color-IIP [6] | 52.1 | 30.2 | 31.7 | 27.8 | 30.2 | 17 | 18 | 14.6 | 12.6 | 37.5 | 27.2 |
| ShapeContext-PS [7] | 81.4 | 67.3 | 59 | 63.9 | 46.3 | 47.3 | 47.8 | 31.2 | 32.1 | 75.6 | 55.2 |
| MultiPart-CBnB | 91.2 | 69.3 | 73.7 | 61.0 | 68.8 | 50.2 | 49.8 | 34.6 | 33.7 | 76.6 | **60.88** |

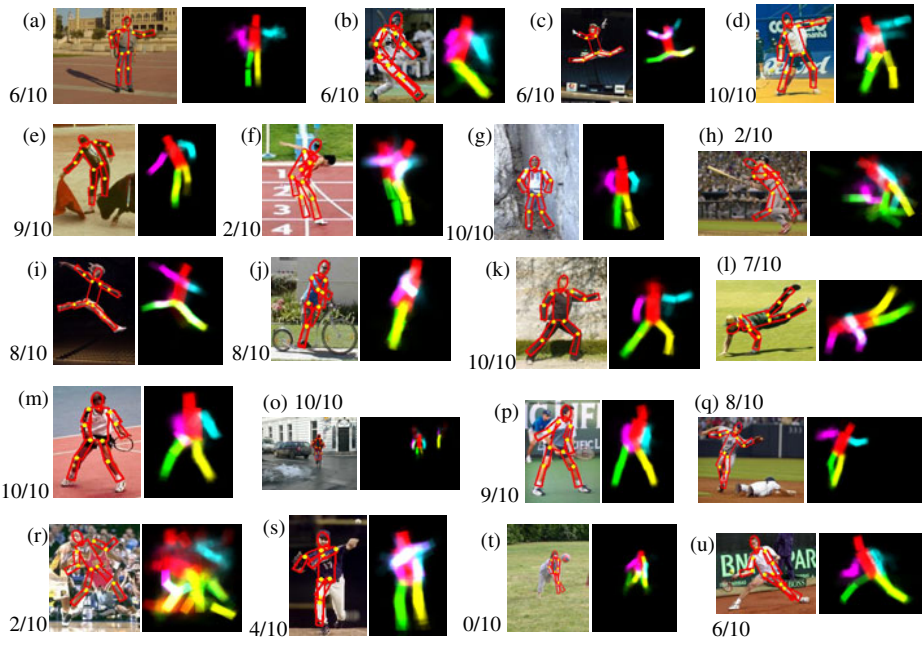


**Fig. 4.** Results on the People Dataset. For each image, the inferred parts (overlaid in red) and the joint likelihood distribution of parts obtained by our algorithm are shown. The image pairs (a-l) are the initial few entries from the dataset (also shown by [7]); (r-u) show some failure cases (see text for description).

Figure 4 shows sample results obtained by our algorithm. Notice that our algorithm performs quite well on fairly articulate poses in cluttered backgrounds 4(d, i, l, m, q). Since our human pose model does not represent non-overlapping and occlusion constraints between parts [8], our algorithm fails to disambiguate parts in such poses 4(f, h, s-u).

**[Comparison with Other Approaches]:** We compare our method with the closely related iterative parsing approach [6] that uses boundary and region templates also used in this work, and pictorial structures with part detectors with boosted shape context features [7]. Table 3 shows the localization accuracy on the People dataset obtained using our approach and those reported by [6], [7]. As Table 3 clearly shows, our method significantly outperforms both methods. A direct speed comparison is difficult as [6], [7] do not report run time numbers.

## 6 Conclusion

In this paper we proposed a part based model with heterogeneous part features for object pose estimation and demonstrated that the use of multiple detectors for each part improves the accuracy. For efficient inference over these models, we created a hierarchical ordering over the different detectors for each part based on accuracy and efficiency and proposed a collaborative branch and bound algorithm which progressively reduces the search space for each part by imposing kinematic constraints from the neighboring parts. We applied our approach to estimate human pose in a single image and outperformed the state-of-art approaches, and showed that our inference is 3 times faster than dense sampling methods using a single, highly accurate detector and 3.7 times faster than those using multiple detectors.

**Acknowledgements.** This research was supported, in part, by the Office of Naval Research under grants #N00014-06-1-0470 and #N00014-10-1-0517. The authors would also like to thank Deva Ramanan for providing the edge and region templates and the People dataset.

## References

1. Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. IJCV 61, 55–79 (2005)
2. Hua, G., Yang, M.H., Wu, Y.: Learning to estimate human pose with data driven belief propagation. In: CVPR, vol. 2, pp. 747–754 (2005)
3. Zhang, J., Luo, J., Collins, R., Liu, Y.: Body localization in still images using hierarchical models and hybrid search. In: CVPR, pp. 1536–1543 (2006)
4. Ramanan, D., Sminchisescu, C.: Training deformable models for localization. In: CVPR, vol. 1, pp. 206–213 (2006)
5. Felzenszwalb, P., Mcallester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: CVPR (2008)

6. Ramanan, D.: Learning to parse images of articulated bodies. In: NIPS, vol. 19, pp. 1129–1136 (2007)
7. Andriluka, M., Roth, S., Schiele, B.: Pictorial structures revisited: People detection and articulated pose estimation. In: CVPR (2009)
8. Wang, Y., Mori, G.: Multiple tree models for occlusion and spatial constraints in human pose estimation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 710–724. Springer, Heidelberg (2008)
9. Sigal, L., Roth, S., Black, M.J., Isard, M.: Tracking loose-limbed people. In: CVPR, pp. 421–428 (2004)
10. Ren, X., Berg, A.C., Malik, J.: Recovering human body configurations using pairwise constraints between parts. In: ICCV, pp. 824–831 (2005)
11. Jiang, H., Martin, D.: Global pose estimation using non-tree models. In: CVPR (2008)
12. Zhu, L., Chen, Y., Lu, Y., Lin, C., Yuille, A.: Max margin and/or graph learning for parsing the human body. In: CVPR (2008)
13. Chen, Y., Zhu, L., Lin, C., Yuille, A., Zhang, H.: Rapid inference on a novel and/or graph for object detection, segmentation and parsing. In: Advances in Neural Information Processing Systems 2008, pp. 289–296 (2008)
14. Lee, M., Nevatia, R.: Human pose tracking using multi-level structured models. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 368–381. Springer, Heidelberg (2006)
15. Ferrari, V., Marin-Jimenez, M., Zisserman, A.: Progressive search space reduction for human pose estimation. In: CVPR (2008)
16. Bourdev, L., Malik, J.: Poselets: Body part detectors trained using 3d human pose annotations. In: ICCV (2009)
17. Collins, M.: Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In: EMNLP (2002)
18. Lampert, C., Blaschko, M., Hofmann, T.: Beyond sliding windows: Object localization by efficient subwindow search. In: CVPR (2008)
19. Kschischang, F., Frey, B., Loeliger, H.A.: Factor graphs and the sum-product algorithm. IEEE Transactions on Information Theory 47, 498–519 (2001)
20. Huang, C., Nevatia, R.: High performance object detection by collaborative learning of joint ranking of granule features. In: CVPR (2010)
21. Dalal, N., Triggs, B.: Histogram of oriented gradients for human detection. In: CVPR, pp. 886–893 (2005)
22. Wu, B., Nevatia, R.: Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In: ICCV, pp. 90–97 (2005)